

CM50109

FORMAL METHODS AND PROGRAMMING

COURSEWORK 2

EngD GROUP 5

DOCUMENT 2

Matthew Thompson (Group Leader)

Milto Miltiadou

Hashim Yaqub

## 1) Testing

### Testing Method

Testing was done very simply: a Test class was created that initialised the objects it needed to perform each test, then ran the tests. Each test was run individually rather than running all tests in a batch, to make sure the program did not crash or produce any unusual errors. Occasionally, random numbers were replaced with fixed values in some classes to provide deterministic tests. All tests done were black box tests: doing white box testing would have taken too much time to justify the effort involved.

### 1. Unit Testing

#### Card, CardManager and Related Classes

Characteristic under test	Test	Expected result	Actual result
Incorrectly formatted card input file	Delete flag (first character) in second line of chance card file	Error message	No message, segmentation fault when landing on chance (FAIL)
Player lands on community chest	Manually move a player to community chest tile. Try a second time to see if it is random.	Card is picked at random, action performed. Second time card is different.	Got two different community chest cards. (PASS)
Player lands on chance	Manually move a player to chance tile. Try a second time to see if it is random.	Card is picked at random, action performed. Second time card is different.	Got two different chance cards (PASS)
Player receives money from card	Manually assign "Won a crossword competition" action.	Player gets £100	Player got £100 (PASS)
Player loses money from card	Manually assign "Drunk in charge" card	Player loses £20	Player lost £20 (PASS)
Player goes to jail	Manually assign "Go to jail" card	Player goes to jail	Player went to jail (PASS)
Player gets moved to a certain position	Manually assign "Advance to go" card	Player goes to "go" square	Player went to "go" square (PASS)
Test every card	Manually do every card's action on player	No bugs or errors	No bugs or errors (PASS)

## Player and PlayerManager Class

Characteristic under test	Test	Expected result	Actual result
Player lands on 'go to jail' tile	Manually move player to “go to jail” tile	Player goes to jail	Player went to jail (PASS)
Player buys a house	Give player all brown properties, land on Old Kent Road, select option to buy house	Player gets house on Old Kent Road, pays £50	Player got house on Old Kent Road, paid £50 (PASS)
Player tries to buy a house, but doesn't have enough money	Set player's balance to 1, try to buy Old Kent Road	Game prints “you can't afford that property” message	Game printed “you can't afford that property” message (PASS)
Player pays rent	Give Old Kent Road to player A, send player B to Old Kent Road	Player B loses £2, player A gets £2	Player B lost £2, player A got £2 (PASS)
Player rolls a double	Manually set dice to 1, 1	Player gets another go	Player got another go (PASS)
Test player landing on every tile	Manually move player to every tile	No bugs or errors	No bugs or errors (PASS)

## Property and Related Classes

Characteristic under test	Test	Expected result	Actual result
Pay rent on a station	Sell station to player A, manually send player B to that station	Player B pays £25 rent	Player B paid £25 rent (PASS)
Pay rent on a utility	Sell utility to player A, manually send player B to that utility	Player B pays four times dice roll	Player B paid four times dice roll (PASS)
Pay rent on 1 house	Give player A a house on Old Kent Road, send player B to Old Kent Road	Player B pays £10 to player A	Player B paid £10 to player A (PASS)

Pay rent on 2 houses	Give player A two houses on Old Kent Road, send player B to Old Kent Road	Player B pays £30 to player A	Player B paid £30 to player A (PASS)
Pay rent on 3 houses	Give player A three houses on Old Kent Road, send player B to Old Kent Road	Player B pays £90 to player A	Player B paid £90 to player A (PASS)
Pay rent on 4 houses	Give player A four houses on Old Kent Road, send player B to Old Kent Road	Player B pays £160 to player A	Player B paid £160 to player A (PASS)
Pay rent on a hotel	Give player A a hotel on Old Kent Road, send player B to Old Kent Road	Player B pays £250 to player A	Player B paid £250 to player A (PASS)

## 2. Integration Testing

### Game and Board Classes

Characteristic under test	Test	Expected result	Actual result
Incorrectly formatted board input file	Remove colour field from Old Kent Road	Error message	Colour was set to "OLD" (FAIL)
Insufficient tiles in board input file	Remove Old Kent Road	Error message	No error message; segmentation fault if last tile on board is reached (FAIL)

### 3. System Testing

#### Functional Testing

Characteristic under test	Test	Expected result	Actual result
Finish game with 2 players	Keep playing the game until it finishes	Game finishes without errors or bugs	Game finished without errors or bugs (PASS)
Finish game with 3 players	Keep playing the game until it finishes	Game finishes without errors or bugs	Game finished without errors or bugs (PASS)
Finish game with 4 players	Keep playing the game until it finishes	Game finishes without errors or bugs	Game finished without errors or bugs (PASS)
Finish game with 5 players	Keep playing the game until it finishes	Game finishes without errors or bugs	Game finished without errors or bugs (PASS)
Finish game with 6 players	Keep playing the game until it finishes	Game finishes without errors or bugs	Game finished without errors or bugs (PASS)

## **2) Limitations and Further Implementation**

In this sections we explain the limitations of our program and what can be done to improve our program in the future. Most of the limitations are features could not implemented due to the limited amount of time.

### **Utility and Railway Stations**

Even though all the relations between the properties are saved, utilities and stations do not used those relations. According to the Monopoly Specifications a player pays more if he lands on a station/utility whose owner owns more than one station/utility. In the game submitted the rent for the stations and utilities is constant. This can be fixed by first checking how many utilities/stations the owner of the current tile owns in the GroupsManager.

'Utilities' was also simplified. Instead of using the total current roll, the player rolls the dices again. This could be fixed using static variables for saving the number of the dices and modifying the 'getTotal()' method of the Dice to static. Another approach is to pass the dices as a parameter, but this will result passing the dices to all the action functions of the all the tiles.

### **Building Houses**

Houses have to be evenly built. In our case, the houses are evenly build but their position is automatically chosen. For example if the player ask to build 1 house on group with three properties and no houses has been build on it yet then the house will be build on the property that was first added on the board.

### **The 'Insufficient Funds' Rules**

If a player does not have enough money to pay then he is withdrawn from the game. In an actual monopoly game the player should have the chance to either sell houses, mortgaged properties or trade his properties to other players. All those are features that can be added to our program.

### 3) Maintenance Guide

#### Editing and Customising Board Tiles

##### **Introduction:-**

For those familiar to the game of Monopoly, they are aware that the board consists of 40 property tiles and special tiles. Each tile has a unique attribute, and some have special cases and attributes. In this implementation of the game, it is possible to players have a bit more fun by creating their own names and groups for properties, as well as set their own prices. There are, however, guidelines which need to be followed in order to successfully create your very own customised monopoly board.

##### **How To Edit Board Tiles:-**

Within the project files of the game, there is a file simply named **'Board'**, which can be viewed and edited in any text editor. There is a list of 40 lines within this file. Each line starts with a number and at least one flag. The numbers represent the position on the board, and the flags determine what kind of card it is. Based on these flags, each line is interpreted by the game differently.

##### **IMPORTANT NOTES BEFORE EDITTING FILE:-**

It must be made clear that not all the flags in this user manual will be covered in detail simply because they **SHOULD NOT** be edited at all unless it's for upgrading the program. For those users who are simply looking to customise the board, **you should not edit anything line the 'o' or 'c' flag, i.e. GO, JAIL, FREE PARKING, GO TO JAIL, SUPER TAX, COMMUNITY CHEST and CHANCE. These tiles are specially reserved tiles which are integral to the game structure. Changing their position on the board is however possible (except for GO), although not recommended.**

##### **Types of Properties and their Flags:-**

###### 1) Normal Property (FLAG: **p n**)

- These make up the majority of tiles in the game. For these you have to indicate The price to purchase, cost of houses and various rent prices.
- The format is as follows (with example):-

Cell Num	flag	Purchase Price	House Price	Base Rent	Rent 1 House	Rent 2 House	Rent 3 House	Rent 4 House	Rent Hotel	Group Colour	NAME
01	p n	60	50	2	10	30	90	160	250	brown	OLD KENT ROAD

###### 2) Railway Stations (FLAG: **p s**)

- By default there are four stations.
- The rent price of the station is dependent on the how many Railway Stations are owned by that player. The format is as follows (with example):-

Cell Num	flag	Purchase Price	Rent 1 Station	Rent 2 Station	Rent 3 Station	Rent 4 Station	NAME
1	p s	200	25	50	100	200	KINGS CROSS STATION

###### 3) Utilities (Flag: **p u**)

- By default there are 2 utilities (Water Works and Electric Company)

- b. The rent price of each utility is dependent on 2 factors
  - i. Number of Utilities owned by the player
  - ii. The value of the SECOND dice roll after a player lands on a utility
- c. Example: If Player 1 owns one utility, and Player 2 lands on it, then Player 2 must roll the dice again. As Player 1 only owns one of the utilities, Player 2 pays Player 1 **four times the value of the second dice roll**. If Player 1 had both utilities, then it would be 10 times the value
- d. The format is as follows (with example):-

Cell Num	flag	Purchase Price	Multiplication Factor 1	Multiplication Factor 2	NAME
12	p u	150	4	10	ELECTRIC COMPANY

## Editing and Customising ‘Community Chest’ and ‘Chance’ Cards

### **Introduction:-**

The ‘Community Chest’ and ‘Chance’ tiles are two very important functions in the game of Monopoly. Conceptually, they both consist of a deck of cards with instructions which directly affect the player who lands one of these tiles. This is a guide to editing and adding your own cards to the deck to further customise the game.

### **How To Edit Board Tiles:-**

Like board, set of cards is stored in ‘**Community Chest**’ and ‘**Chance**’ file, which can be viewed and edited in any text editor. Unlike Board, there is no particular limit on the number of cards in each of the files, and every card can be edited, as long as it conforms to guidelines. Thi

### **IMPORTANT NOTES BEFORE EDITTING FILE:-**

Any card with the flag ‘**f**’ or ‘**j**’ (‘Get Out of Jail Free’ and ‘Go To Jail respectively’) can be removed from the deck. However they should not be edited.

### **Types of Cards and their Flags:-**

Each card takes 0, 1 or two parameters. The general rule is FLAG, PARAMETERS, INSTRUCTION.

- 1) Binary Choice or ‘Lose Money or Take a Chance’ (Flag: **b**)
  - a. This type of card gives the player a decision between paying a fee to the bank, or using a ‘Chance’ card instead.
  - b. This card takes **1 parameter**, which is the amount of money to **deduct from the player’s balance**.
  - c. **Example:** b **10** Pay £10 or Take a Chance
- 2) Player Receives Money (Flag: **g**)
  - a. This type of card allows the player to receive a specified amount of money
  - b. This card takes **1 parameter**, which is the amount of money to **add to the player’s balance**.
  - c. **Example:** g **20** Income Tax Refund. Collect £20



- 3) Player Receives Money From Other Players (Flag: **gp**)
  - a. This type of card allows the player to receive a specified amount of money from all of the other players
  - b. This card takes **1 parameter**, which is the amount of **money, multiplied by number of other players, to add to the player's balance**, and also the money to be **deducted from all other players balance**.
  - c. **Example:** gp **10** It's your Birthday. Collect £10 from each Player
- 4) Repairs Costs for Houses and Hotels (Flag: **h**)
  - a. This type of card removes money from the player. The amount of money to remove is dependent on the number of houses and hotels the player owns.
  - b. This card takes **2 parameter**, which are the **cost of each house**, and the **cost of each hotel**.
  - c. **Example:** h **40 115** You Are Assessed for Street Repairs. £40 per House, £115 per Hotel
- 5) Player Loses Money (Flag: **l**)
  - a. This type of card allows the player to lose a specified amount of money
  - b. This card takes **1 parameter**, which is the amount of money to **deduct from the player's balance**.
  - c. **Example:** l **20** "Drunk in Charge" fine £20
- 6) Move Player to Position (Flag: **m**)
  - a. This type of card allows the player to move to a specific location on the board, using the integer position as a reference.
  - b. This card takes **1 parameter**, which the **position on the board** the player should be moved to.
  - c. **Example:** m **0** Advance to Go
- 7) Move Player Back a Certain Amount (Flag: **mb**)
  - a. This type of card allows the player to move back a certain amount of tiles
  - b. This card takes **1 parameter**, which the is the **number of tiles a player moves back**.
  - c. **Example:** mb **3** Move Back 3 Spaces

## 4) IMPORTANT – CODE IMPLEMENTATION

The code implementation is located in 'listings.pdf'

Full documentation for code was generated using 'Doxygen' and is located in the 'Doxygen' Folder